SD6909.1/S97553
September 12, 2003

CLEARANCE DETECTOR AND METHOD FOR MOTION AND DISTANCE

INVENTOR:

Patrick G. Xavier
9917 Fostoria Rd., NE
Albuquerque, NM 87111

T

Express Mail No.: EV332388196US

PATENT APPLICATION

CLEARANCE DETECTOR AND METHOD FOR MOTION AND DISTANCE

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of the filing of U.S. Provisional Patent Application Serial No. 60/410,580, entitled "A Collision Detector and Method for Motion and Distance", filed on September 12, 2002, and the specification thereof is incorporated herein by reference.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

This invention was made with Government support under Contract DE-AC04-94AL85000 awarded by the U.S. Department of Energy. The Government has certain rights in the invention.

INCORPORATION BY REFERENCE OF MATERIAL SUBMITTED ON A COMPACT DISC

Not Applicable.

COPYRIGHTED MATERIAL

Not Applicable.

BACKGROUND OF THE INVENTION

Field of the Invention (Technical Field):

This invention relates to the field of modeling interactions among entities having geometry, specifically for interference and collision detection and distance computation.

Description of Related Art:

Note that the following discussion refers to a number of publications by author(s) and year of publication, and that due to recent publication dates certain publications are not to be considered as prior art vis-a-vis the present invention. Discussion of such publications herein is given for more

complete background and is not to be construed as an admission that such publications are prior art for patentability determination purposes.

A recurrent problem in computer-based simulations of geometric volumes is determining the clearances between volumes. Many applications, including robotics and visualization, require modeling of interactions of objects. In robotics, for example, this modeling permits detection of possible collisions between a robot and its environment. Both accuracy and efficiency are needed. Accuracy is needed because inaccurate determination of clearances could lead to physical collisions. Efficiency is needed because, in the real-time computational context of robotics, information about clearances is required while the robot and elements of the environment are in motion.

Early methods for clearance determination did not provide distance information for clearances among volumes. See, e.g., Fuchs et al., "On visible surface generation by a priori tree structures," Proc. of ACM SIGGRAPH, pages 124-133, 1980. Later methods addressed this problem but did not handle swept-body distance or swept-body interference detection. See, e.g., Sato et al., "Efficient collision detection using fast distance calculation algorithms for convex and non-convex objects," Proc. 1996 IEEE Int'l Conf on Robotics and Automation, pages 771-778, Minneapolis, Min., April 1996; Gottschalk et al., "OBB-tree: A hierarchical structure for rapid interference detection," Proc ACM SIGGRAPH'96, pages 171-180, August 1996. Further methods were able to handle swept volumes but did not provide clearance distance information for volumes that did not collide. See Canny, John, "Collision detection for moving polyhedra," IEEE Trans. on Pattern Analysis and Machine Intelligence, 8(2):200-209, 1986; and Cameron, "Collision detection by 4D intersection testing," Int'l Journal of Robotics Research, 6(3):291-302, June 1990.

These drawbacks were substantially addressed by Applicant in U.S. Patent No. 6,099,573, Method and apparatus for modeling interactions; and U.S. Patent No. 6,407,748, Method and apparatus for modeling interactions. These advances are discussed in Xavier, Patrick, G., Implicit Convex-Hull Distance of Finite-Screw-Swept Volumes, 2002 IEEE International Conference on Robotics and

Automation (ICRA) (May 2002). U. S. Patent No. 6,099,573 involved a method for determining collisions and clearances, including distances, through calculations of bounding volumes based on unions of convex polygons and polyhedra. This method, which computes an approximate minimum clearance between the volumes swept by two bodies under motion, was faster and more robust than the prior art. U.S. Patent No. 6,407,748 involved a complementary computation of clearances where two bodies were undergoing translation. This method computes the exact minimum clearance (to numerical precision limited by the representation of floating point numbers on the computer component of the invention) between the volumes swept by two bodies only undergoing translation.

Performance of applications such as robots and visualization is still limited by the computational efficiency of the clearance calculations. U.S. Patent No. 6,099,573, and U.S. Patent No. 6,407,748 both required explicit calculation of all the vertices of polyhedral approximations of the convex hulls of volumes swept by various nodes in the representations of the modeled objects. The present invention provides a more efficient computation of the clearance calculation by adapting an algorithm of Gilbert and Foo (Gilbert and Foo, Computing the Distance Between General Convex Objects in Three-Dimensional Space, IEEE Transactions on Robotics and Automation, Vol. 6, No. 1, February 1990, pages 53-61) thereby obviating the need to calculate explicit representations (exact or approximate) of convex hulls of volumes swept by various nodes in the representations of the modeled objects. The prior art was also limited by the kinds of the volumes that could be modeled. The present invention further permits use of a greater number of geometric primitives in modeling the volumes, including cones, cylinders and discs.

By avoiding computation of any (approximate) representations of the volumes swept by sections of the modeled bodies, the present invention comprises a method that is faster than prior art. The hierarchical distance implementation of the present invention does not construct a swept bounding volume hierarchy (BVH) representation of either of the first or the second bodies. Rather, once there is an initial BVH for the first body and a BVH for the second body each in a canonical (i.e., "home") position and orientation without any sweeping, it avoids constructing any other BVH or BVH node for

either body. The bodies are represented by unions of Convex Geometric Primitives (CGPs), including but not limited to convex polygons, convex polyhedra, closed discs, cones, cylinders, and spheres. From these CGPs, a bounding volume hierarchy (BVH) representation is generated in pre-processing each body. This can be done using various methods known to those skilled in the art. A BVH is a tree whose nodes are each CGPs, and whose leaves are the original CGPs from the body in the "home" position and orientation.

A key advance of the present invention is that it does not compute swept volumes. Instead, for a series of pairs of nodes, one node each from the respective BVHs, it determines the clearance between the convex hulls of the volumes "virtually" swept by the two nodes in the pair under the motions of their respective bodies. This makes the present invention more accurate than U.S. Patent No. 6,099,573. Towards determining this clearance, the method determines for each of a series of direction vectors the furthest point "virtually" swept by a node (which is a CGP) over a period of time. To determine this clearance, one embodiment of the present invention adapts the algorithm of Gilbert and Foo for finding the minimum distance between two convex geometric objects as the method of determining this clearance by way of determining for each of a series of direction vectors the furthest point "virtually" swept by a node over a period of time.

The method of the present invention determines a representation (also referred to as a canonical representation) of a body in its canonical ("home") position and orientation. A node from this representation that has been mapped to an arbitrary position and orientation can be compared to a direction vector (w-vector or its inverse) to locate the point furthest on that node along the direction vector. To make the description clearer, subsequently, the location (or vertex or point) furthest along a direction vector may be referred to as the directionally furthest location (or vertex or point).

The method is further applied to bodies undergoing a motion, including but not limited to translation, rotation, and translation concurrent with rotation, described by a time-dependent mapping. By determining an inverse mapping of the motion and applying the inverse mapping to the direction

vector (w-vector or its inverse), then mapping the result to the original canonical representation of the body without actually sweeping out the volume, the invention can determine the furthest point along the direction vector (w-vector or its inverse) on the convex hull of the virtually swept volume generated by any given node of the BVH. This method accounts for the swept volume due to the node and its body's motion over a period of time to determine the furthest point in a particular direction during the entire time period and locates this furthest point without performing the brute-force and computationally intensive calculation of a representation of an approximate superset of the volume swept by the node over the period of time. Coordinated computations enable determination of the separation between the volumes virtually swept by pairs of nodes from the respective BVH representations. This determines quick and reliably conservative calculation of collision detection of moving bodies over time to permit collision avoidance.

Where a node comprises a convex polyhedron, the present invention determines the transition of the directionally furthest vertex to an edge-adjacent vertex on that polyhedron due to the motion of the direction vector under an inverse mapping. The determining of transitions from a vertex to an edge-adjacent vertex due to the motion of the direction vector is a unique aspect of the present invention.

Other patents have covered trajectory calculations but do not provide the advance of the present invention. U.S. Patent No. 5,537,119, Method and system for tracking multiple regional objects by multi-dimensional relation, provides a novel algorithm for detecting trajectory collisions, but the method involves tracks of objects that are not modeled with respect to volume. U.S. Patent No. 6,285,805, System and method for finding the distance from a moving query point to the closest point on one or more convex or non-convex shapes, involves a kind of calculation unrelated to the present invention. The method of U.S. Patent No. 6,285,805 is appropriate for polygonal mesh-type models rather than the volume-based models of the present invention. U.S. Patent No. 5,675,720, Method of searching for points of closest approach, and preprocessing method therefore (Sato, et al.), also uses a mesh-type model and uses the original Gilbert algorithm directly. Moreover, although Sato's claims use the term "points of closest approach" in its Background and Description, it is clear that this term refers to the

points of minimum distance between two objects at discrete (separated) points in time, rather than for the modeled objects across the time interval of their motion.

Another approach is to specialize the modeling of the volumes to a particular application domain, so that the collision-detection problem can be solved for a specific application rather than generally. An example of this approach is U.S. Patent No. 6,577,925, Apparatus and method of distributed object handling. In this case, the volumes consist of "envelopes" that have been predetermined, based on known object geometry and trajectories, *before the real-time computation* to ensure non-collisions during subsequent movements along the predetermined trajectories. The clearance-detection computation can the use a brute-force algorithm rather than the more efficient algorithm of the present invention.

Other related work includes inventions for computer animation and virtual reality. Examples include U.S. Patent No. 6,535,215, Method for animating 3-D computer-generated characters, and U.S. Patent No. 6,054,991, Method of modeling player position and movement in a virtual reality system. While animation and VR need collision detection for modeled objects, In both of these cases the underlying representations differ from that of the present invention and hence lead to substantially different algorithms. The animation system, because it is not real-time, uses a brute-force algorithm for collision detection. The VR system, although real-time, does not have true 3-D representation of objects. Rather, it represents the objects in a series of planes, detecting 2-D collisions within the planes, again using a simpler brute-force approach.

BRIEF SUMMARY OF THE INVENTION

The present invention is of software and a method for clearance detection that maps motions of a first body and a second body for detecting a clearance between the first body and the second body, wherein the first body undergoes a first motion and the second body undergoes a second motion, comprising: employing a first representation for the first body; employing a second representation for the second body; employing a first mapping of the first motion, the first mapping having a first inverse

mapping; employing a second mapping of the second motion and having a second inverse mapping; employing a hierarchical minimum distance search with respect to the volumes virtually swept by the bounding volume hierarchy (BVH) representations of the first and second body under their respective motions; applying the first inverse mapping to a first direction vector and the first representation to determine a first directionally furthest location on a first volume virtually swept by a node in the BVH representation of the first body during the first motion; applying the second inverse mapping to a second direction vector and the second representation to determine a second directionally furthest location on a second volume virtually swept by a node in the BVH representation of the second body during the second motion; and detecting a clearance between the first body and the second body. In the preferred embodiment, the invention further comprises determining a size of the clearance and determining collisions. The first and second motions each may be either a translation or a rotation. Determining a size of the clearance preferably comprises: determining a first point on the first representation representing a shortest distance; determining a second point on the second representation representing the shortest distance; and calculating the size of the clearance. Determining a collision preferably comprises: determining a first point on the first representation representing a shortest distance; determining a second point on the second representation representing the shortest distance; and calculating the size of the clearance.

Applying the first inverse mapping to a direction vector and the first representation to determine a first directionally furthest location on a first volume virtually swept during the first motion comprises: determining a first BVH representation from the first representation, wherein the first BVH representation comprises a first tree of nodes, each of which is a convex geometric primitive; applying the first inverse mapping to that direction vector; and determining a directionally furthest location on the first volume virtually swept by one of the one or more of the geometric primitives of the first tree of nodes during the first motion using the first inverse mapping. Applying the second inverse mapping to a direction vector and the second representation to determine a second directionally furthest location on a second volume virtually swept during the second motion comprises: determining a second BVH representation from the second representation, wherein the second BVH representation comprises a second tree of nodes,

each of which is a convex geometric primitive; applying the second inverse mapping to that direction vector; and determining a second directionally furthest location on the second volume virtually swept by one of the convex geometric primitives of the second tree of nodes during the second motion using the second inverse mapping. Detecting a clearance between the first body and the second body comprises: computing a distance between a first convex hull of the first volume virtually swept by one of the convex geometric primitives of the first tree of nodes during the first motion and a second convex hull of the second volume virtually swept by one of the convex geometric primitives of the second tree of nodes during the second motion and repeating the computing step for another of the convex geometric primitives of the first tree of nodes and another of the convex geometric primitives of the second tree of nodes. Detecting a clearance size between the first body and the second body comprises: computing a distance between a first convex hull of the first volume virtually swept by one of the one or more first convex geometric primitives during the first motion and a second convex hull of the second volume virtually swept by one of the one or more second convex geometric primitives during the second motion; repeating the computing step for another of the one or more first convex geometric primitives and another of the one or more second convex geometric primitives; and determining a shortest distance of the distances to determine the clearance size. One or more of the convex geometric primitives may be a convex polyhedra, with the invention additionally comprising: determining the directionally furthest point of a polyhedron's virtually swept volume by determining a directionally furthest vertex on that polyhedron with respect to an initial image of the path under the inverse mapping; determining a transition of the directionally furthest vertex to an edge-adjacent vertex on that polyhedron due to motion of the inversely mapped path; and determining subsequent changes of the directionally furthest vertex to an edge-adjacent vertex on that polyhedron due to motion of the inversely mapped path.

A primary object of the present invention is the correct and efficient detection of clearances between three-dimensional bodies in computer-based simulations, where one or both of the bodies is subject to translation and/or rotary motion. Other objects of the present invention are the correct and efficient determination of the size of such clearances, and the correct and efficient determination of whether there is a collision between the bodies.

A primary advantage of the present invention is that is more efficient and accurate than prior methods. This is particularly important for simulations of moving volumes, which are computationally intensive.

Other objects, advantages and novel features, and further scope of applicability of the present invention will be set forth in part in the detailed description to follow, and in part will become apparent to those skilled in the art upon examination of the following, or may be learned by practice of the invention. The objects and advantages of the invention may be realized and attained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

Not Applicable.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to a method for modeling interactions among entities having geometry, specifically for interference and collision detection and distance computation. Using an adaptation of the Gilbert and Foo algorithm, the method conservatively determines distances between bodies under motion without having to calculate swept volumes. "Conservatively" means that the approximate distance determined by the method will never be smaller than the actual minimum distances between the bodies under motion. The method further uses an adaptation of hierarchical search with respect to bounding volume hierarchies for overall efficiency while maintaining correctness.

Given two bodies, each of which comprises a union of convex geometric primitives and each of which is undergoing separate motions, the method creates representations for the two bodies and mappings and inverse mappings for the motions of the two bodies. The motions of the bodies can begin at positions and orientations other than the bodies' canonical positions and orientations. The method uses the representations, mappings, a hierarchical search, and re-computations of a direction vector to

determine a first convex geometric primitive component from the first body and second convex geometric primitive component from the second body such that the clearance between the convex hull of the volume virtually swept by the first convex primitive under the motion of the first body and the convex hull of the volume virtually swept by the second convex primitive under the motion of the second body conservatively approximates the minimum clearance between the two bodies under their respective motions. A first convex primitive component and a second convex primitive component having this property are known as *witness primitives*. The method either determines that the clearance between the convex hulls of the virtually swept first and second witness primitives is zero or, if the clearance is greater than zero, finds a point on each of these two convex hulls such that the distance between those points is the clearance between these convex hulls. A pair of such points are referred to as *witness points.*

Based on the determination of the distance between the witness points, the size of a positive clearance between the bodies can be calculated. Based on determination that the clearance between the convex hulls of virtually swept volumes of the witness primitives is zero, a zero clearance between the bodies can be determined.

Based on the size of the clearance between the bodies, it can be determined whether the bodies collide. A zero clearance signifies a collision, and a positive clearance robustly signifies no collision.

The clearance-detection method can be applied to bodies where the first body has a screw motion consisting of a translation concurrent with a rotation and the second body has a second screw motion consisting of a second translation concurrent with a second rotation, to bodies where the first body has a translation and the second body has a second translation, to bodies where the first body has a rotation and the second body has a second rotation, and to bodies where one body has a rotation and the other body has a translation.

The clearance-detection method can be applied to time-dependent simulations where the two bodies have series of motions.

The determination of a pair of witness primitives for a pair of bodies undergoing motion can be implemented by determining a bounding volume hierarchy (BVH) representation for each body from its initial representation as a union of convex geometric primitives and then applying an extension of a hierarchical minimum-distance search algorithm to the two BVH representations. The convex primitive components whose union composes a body are the leaf nodes of a BVH representation of that body. Note that the terms "hierarchical bounding volume representation" and "bounding volume hierarchy representation" mean the same thing. At each step of the hierarchical minimum-distance search, instead of simply computing the distance between a node from the BVH representation of the first body and a node from the BVH representation of the second body, the method computes the distance between the convex hulls of the volumes virtually swept by a node from the BVH representation of the first body under its motion and a node from the BVH representation of the second body under its motion. Based on a comparison of this distance with the minimum distance found so far between convex hulls of swept volumes of convex geometric primitive components that composes the first body and convex hulls of swept volumes of convex geometric primitive components that composes the second body, the search algorithm decides whether some subsequent steps in the search will need to consider the swept volume of one of the two nodes with respect to the swept volumes of each of the children of the other. The pair of convex primitive components with the distance between the convex hulls of their virtually swept volumes that is the minimum of those found when the search terminates is the pair of witness primitives.

The clearance between the bodies can be determined by computing the distance between the convex hulls of the volumes virtually swept by the witness primitives found by this extension to swept-volume hierarchical minimum-distance search.

For purposes of calculating the distance between the convex hulls of the volumes virtually swept by a pair of nodes from the respective BVH representations of the two bodies, an adaptation of the Gilbert-Foo algorithm for computing the distance between two convex geometric objects is used. The key computation that the present invention adapts is that of determining a furthest location on a convex geometric object. The determination of the directionally furthest location of the convex hull of the volume virtually swept by the node from the BVH representation of the first body can be implemented by applying the relevant inverse mapping to the direction vector; and then using the inverse mapping to find the directionally furthest location on that node with respect to the (moving) inversely-mapped vector. The determination of the directionally furthest location of the convex hull of the volume virtually swept by the node from the BVH representation of the second body can be implemented using the BVH representation, mapping, vector and motion appropriate to the second body.

The convex geometric primitives composing the two bodies are selected from the group consisting of polygons, polyhedra, cylinders, cones, and other closed geometry.

The clearance-detection method for bodies comprising convex geometric primitives can be applied to time-dependent simulations where the two bodies have series of motions.

The method for clearance detection can be applied to the case where one body undergoes a motion and the other body is at rest. In this case, the method creates BVH representations for the two bodies and creates a mapping and inverse mapping for the motion of the moving body. The motion of the moving body can begin at a position and orientation other than the body's canonical position and orientation. The method uses the representations, mappings, hierarchical search, and re-computations of a direction vector to determine a convex geometric primitive component from the moving body and a convex geometric primitive component from the stationary body such that the clearance between the convex hull of the volume virtually swept by the convex primitive from the moving body under the motion of that body and the convex primitive from the stationary body conservatively approximates the clearance between the two bodies under their respective motions. Such a pair of convex primitive

components from the respective bodies and having this property are known as *witness primitives*. The method either determines that the clearance between the stationary witness primitive and the convex hull of the virtually swept volume of the moving witness primitive is zero, or , if the clearance is greater than zero, finds a point on the stationary witness primitive and a point on the convex hull of the virtually swept volume of the moving witness primitive such that the distance between those points is the clearance between the convex hull of volume virtually swept by the moving witness primitive and the stationary convex primitive. A pair of such points are referred to as *witness points.*

Where one body is moving and the other body is at rest, for purposes of calculating the distance between a node from the BVH representation of the stationary body and the convex hull of the volume virtually swept by a node from the BVH representations of the moving bodies, an extension to the Gilbert-Foo algorithm for computing the distance between two convex geometric objects is used. The determination of the directionally furthest location of the convex hull of the volume virtually swept by the node from the BVH representation of the moving body can be implemented by applying the relevant inverse mapping to the direction vector; and then using the inverse mapping to find the directionally furthest location on that node with respect to the (moving) inversely-mapped vector.

Where one body is moving and the other body is at rest, the clearance-detection method can be applied to bodies where the moving body has a screw motion consisting of a translation concurrent with a rotation, to bodies where the moving body has a translation, and to bodies where the moving body has a rotation.

Where a node in the BVH representation of a body is convex polyhedron and that body is moving, the determination of the directionally furthest location on the convex hull of the volume virtually swept by that node under the motion of the body can be implemented by applying the relevant inverse mapping to the direction vector; and then using the inverse mapping to find the directionally furthest location on that node with respect to the (moving) inversely-mapped vector. The method uses the polyhedron representation, inverse mapping, and vector to determine the directionally furthest vertex of

the polyhedron, transitions of the directionally furthest vertex to an edge-adjacent vertex on that polyhedron due to motion of the inversely mapped direction vector, and subsequent changes of the directionally furthest vertex to an edge-adjacent vertex due to motion of the inversely mapped direction vector.

A more detailed description of the invention in its preferred form is next provided, employing the following definitions of interference detection, distance computation, and collision detection for a single moving body. Interference detection (or clash detection) for a rigid body $\mathcal{R}$ at a position and orientation $\mathbf{x}$ and obstacles $\mathcal{O}$ means to determine whether $\mathcal{R}(\mathbf{x}) \cap \mathcal{O}$ is non-empty, with $\mathcal{R}(\mathbf{x})$ denoting the image of $\mathcal{R}$ in the world. Distance computation means to determine the minimum distance between all points in $\mathcal{R}(\mathbf{x})$ and those in $\mathcal{O}$. Simple collision detection for a rigid body $\mathcal{R}$ moving over a path segment among obstacles $\mathcal{O}$ means to determine whether at any point along the path the body contacts or intersects any obstacles. In other words, if one lets C denote the space of positions and orientations and let $\mathbf{p} : [0,1] \rightarrow C$ denote the path segment, then collision detection asks whether there is any t in [0,1] such that $\mathcal{R}(\mathbf{p}(t)) \cap \mathcal{O}$ is non-empty.

This application first reviews the Gilbert/Foo algorithm at a high level, where the original, generalized, and enhanced versions are identical. It then conceptually describes how to extend the Gilbert/Foo algorithm to compute the distance between the convex hulls of the volumes swept by a pair convex geometric primitives undergoing finite screw motion. It then reviews the use of bounding volume hierarchies and convex geometric primitives for distance computation between rigid bodies and describe the extension that yields the general result.

Given compact set $\mathcal{X} \subset \mathrm{R}^k$, define support functions h_X and s_X such that for each nonzero vector $\mathbf{w} \in \mathrm{R}^K$

$$h_X(\mathbf{w}) = \max_{\mathbf{x} \in \mathcal{X}} \mathbf{w} \cdot \mathbf{x}; \tag{1}$$

$$s_X(\mathbf{w}) = \text{some } \mathbf{x} \in \mathcal{X} \text{ s.t. } \mathbf{w} \cdot \mathbf{x} = h_X(\mathbf{w}). \tag{2}$$

If the support functions for $X_1, X_2 \subset \mathrm{R}^k$ can be computed, then one can apply the generalized Gilbert/Foo algorithm to find the minimum distance vector between their convex hulls *conv*(X_1) and *conv*(X_2). In the simple case, X_1 are X_2 are convex and identical to their convex hulls.

Pseudocode for the Gilbert/Foo algorithm is presented below. The key observation is that the only operations that are directly dependent on the inputs *A* and *B* are the support functions s_A and s_B and finding initial points in *A* and *B*. *closestToOrigin*(*S*) finds the point in *S* closest to the origin. *minimalSimplex*(*S*,**w**) finds the minimal simplex generated by the vertices of *S* that contains **w**. *terminationCondition*(**w**,**x**,*dist*) is a predicate that decides whether **w** and *dist* are as close to the actual minimum distance vector and distance as can be computed by the algorithm; it returns true when *dist* is zero, which is the case when *S* has full dimension.

real **Generalized Gilbert/Foo**(*convex A, convex B*)

{

$\mathbf{x}_A \in A; \mathbf{x}_B \in B$

$\mathbf{w} = \mathbf{x} = \mathbf{x}_B - \mathbf{x}_A;$

simplex $S = \{\mathbf{x}\};$

do {

$\mathbf{x}_A = s_A(\mathbf{w}); \mathbf{x}_B = s_B(-\mathbf{w});$

$\mathbf{x} = \mathbf{x}_B - \mathbf{x}_A;$

S =simplex($S \cup \{\mathbf{x}\}$);

w =closestToOrigin(*S*);

S =mimimalSimplex(*S*,**w**);

dist = $\sqrt{\mathbf{w} \cdot \mathbf{w}}$

} while (!terminationCondition(**w**, **x**, dist);

return (**w**, dist);

}

The enhanced Gilbert/Foo Algorithm for convex polyhedra improves on the original by using hill-climbing in computing the support functions. This lowers the complexity of the loop interior to roughly $O(\sqrt{n})$ from $O(n)$. Applicant was previously unable to apply hill-climbing to Applicant's earlier conservative distance algorithm because for a given polyhedron's vertex set it generated points that could lie interior to the convex hull. In contrast, the method of the present invention is roughly $O(\sqrt{n})$ and applies hill-climbing.

To extend the generalized Gilbert/Foo algorithm to the convex hulls of the volumes swept by a pair of convex geometric primitives (CGPs), one just needs to compute the support functions of the convex hulls of the swept volumes. For example, one observes that the convex hull of the vertices of a polyhedron and their (swept) images under any given motion is equal to the convex hull of the volume swept by that polyhedron under that motion. Therefore, if it is possible to compute $h_X(.)$ and $s_X(.)$ whenever X is the image of the vertices of a polyhedron under a given class of motions, the distance between the convex hulls of the volumes swept out by any two polyhedra undergoing motions from this class can be computed by applying the generalized Gilbert/Foo algorithm. The complexity of the loop interior remains roughly $O(\sqrt{n})$ when the rotation angle is bounded by a constant, e.g., 6.2832.

Next presented is a general approach to computing support functions for CGPs undergoing finite screw motions below. Further below is shown in detail how $h_X(.)$ and $s_X(.)$ can be computed when X is the image of a point under finite screw motion $\mathbf{X} = (\hat{\mathbf{n}}, \theta, \mathbf{x}_a, \mathbf{x}_{tr})$, which is given by

$$\mathbf{X}(t,\mathbf{p}) = \mathbf{R}(\hat{\mathbf{n}}, t\theta)(\mathbf{p} - \mathbf{x}_a) + \mathbf{x}_a + t\mathbf{x}_{tr}, \tag{3}$$

where $\mathbf{p}$ is the point being moved, path parameter $t \in [0,1]$, $\mathbf{x}_{tr}$ is the translational component, $\mathbf{R}(\hat{\mathbf{n}}, \alpha)$ is the rotation matrix for non-negative angle α about $\hat{\mathbf{n}}$, and the original axis of rotation goes through $\mathbf{x}_a$. Below, the invention extends this to compute $h_X(.)$ and $s_X(.)$ when X is the image of a convex

polyhedron (or polygon) undergoing finite screw motion. The general approach below can be applied to other CGPs such as discs, cones, cylinders, etc.

This application has so far used static compact sets $X \subset \mathrm{R}^k$ to index the support functions, but one now wishes to consider sets in R^k that move according to screw motion. Overloading the notation, functions *s*(..) and *h*(..) are defined such that for all $\mathbf{w} \in \mathrm{R}^k$ and compact subsets X of R^k

$$s(\mathbf{w}, X) = s_X(\mathbf{w}); \tag{4}$$

$$h(\mathbf{w}, X) = h_X(\mathbf{w}). \tag{5}$$

Let $\mathbf{X} = (\hat{\mathbf{n}}, \theta, \mathbf{x}_a, \mathbf{x}_{tr})$ be a screw motion as above in (3). Further overloading the notation, define

$$\mathbf{X}(t, X) = \{\mathbf{x} : \mathbf{x} = \mathbf{X}(t, \mathbf{y}) \text{ for some } \mathbf{y} \in X\}; \tag{6}$$

$$\mathbf{X}(X) = \{\mathbf{x} \in \mathbf{X}(t, X) \text{ for some } t \in [0,1]. \tag{7}$$

For any such given X and $\mathbf{X}$, clearly

$$h(\mathbf{w}, conv(\mathbf{X}(X))) = h(\mathbf{w}, \mathbf{X}(X)). \tag{8}$$

It follows that if X and Y are compact and $\mathbf{X}$ and $\mathbf{Y}$ are finite screw motions, and one can compute $s(\mathbf{w},\mathbf{X}(X))$, $h(\mathbf{w},\mathbf{X}(X))$, $s(\mathbf{w},\mathbf{Y}(Y))$, and $h(\mathbf{w},\mathbf{Y}(Y))$, for all $\mathbf{w} \in \mathrm{R}^k$, then one can apply the generalized Gilbert/Foo algorithm to compute the minimum distance and a minimum distance vector between $conv(\mathbf{X}(X))$ and $conv(\mathbf{Y}(Y))$.

Given $\mathbf{X} = (\hat{\mathbf{n}}, \theta, \mathbf{x}_a, \mathbf{x}_{tr})$, one has

$$h(\mathbf{w}, \mathbf{X}(t, X)) = h(\mathbf{R}^T(\hat{\mathbf{n}}, t\theta)\mathbf{w}, X) - (\mathbf{R}(\hat{\mathbf{n}}, t\theta)\mathbf{x}_a) \cdot \mathbf{w} + \mathbf{x}_a \cdot \mathbf{w} + t\mathbf{x}_{tr} \cdot \mathbf{w}; \tag{9}$$

$$s(\mathbf{w}, \mathbf{X}(t, X)) = \mathbf{R}(\hat{\mathbf{n}}, t\theta)\{s(\mathbf{R}^T(\hat{\mathbf{n}}, t\theta)\mathbf{w}, X)\} - \mathbf{R}(\hat{\mathbf{n}}, t\theta)\mathbf{x}_a + \mathbf{x}_a + t\mathbf{x}_{tr}. \tag{10}$$

More generally, suppose that $\mathcal{P}$ is the convex hull of X. Then the convex hull of the volume swept by $\mathcal{P}$ under a time-dependent, convexity-preserving map $\mathbf{X}$ is the convex hull of the volume swept by X under that map. Suppose that $\mathcal{Q}$ is the convex hull of set Y and $\mathbf{Y}$ is a time dependent, convexity-preserving map. If one can compute the support functions $h(..)$ and $s(..)$ for $\mathbf{X}(X)$ and $\mathbf{Y}(Y)$, then one can compute the distance and distance vector between the convex hulls of the volumes swept by $\mathcal{P}$ and $\mathcal{Q}$ under motions $\mathbf{X}$ and $\mathbf{Y}$. For example, if one can compute the support functions of discs and points under finite screw motion, the class of geometric objects for which one can do this includes cones and cylinders.

Since

$$h(\mathbf{w},\mathbf{X}(X)) = \max_{t\in[0,1]} h(\mathbf{w},\mathbf{X}(t,X)), \qquad (11)$$

computing $h(\mathbf{w},\mathbf{X}(X))$ and $s(\mathbf{w},\mathbf{X}(X))$ reduces to finding some $t^* \in [0,1]$ that maximizes $h(\mathbf{w},\mathbf{X}(t,X))$. To find maxima in (0, 1), one computes time derivatives of $h(\mathbf{w},\mathbf{X}(t,X))$ in (9) and solves for t that zeroes the first derivative. If X is a point, disc, or ellipse, then this condition can be reduced to solving an equation that is algebraic in $\cos(t\theta)$, so that substituting in $y = \cos(t\theta)$ yields an equation that is algebraic in y. The usual methods can then be used to find $\max_{t\in[0,1]} h(\mathbf{w},\mathbf{X}(t,X))$. Plugging the maximizing t into (9) and (10), one obtains the values of the support functions. It follows that one can compute the support functions of the volumes swept by the convex hulls of finite sets of points, discs, and ellipses under screw motion.

Note that the methods for computing the support functions of the volumes swept by CGPs under finite screw motion can be used to compute tight bounding boxes or k-DOPs around those swept volumes; the former is preferred.

Assume that bodies are represented by unions of Convex Geometric Primitives (CGPs) with unrestricted orientation, for example: convex polygons, convex polyhedra, closed discs, cones, cylinders or spheres. From these CGPs, in pre-processing one generates a bounding volume hierarchy (BVH) representation of the body, which can be done using various known methods. A BVH is a tree whose nodes are each CGPs, and whose leaves are the original CGPs from the model. The subtree rooted at a node represents the union of the primitives at its leaves, and each non-leaf node is a spatial superset of the object represented by its subtree.

Given a BVH representation of a body, it is simple to obtain a BVH that results from applying a rigid-body transformation to it. The tree structures are identical. One applies the transformation to map each node in the original BVH to a corresponding node in the mapped BVH. In the preferred embodiment, one does not actually construct the mapped BVHs but computes distance computations between mapped nodes as they are needed by incorporating the transformations into CGP-CGP distance code.

```
real BVH_Dist(body *BODY1, body *BODY2)
{
        real dist ← ∞;
        pairQueue queue;
        body *b1,*b2;
        queue.enqueue(BODY1, BODY2, 0);
        while (!queue.isEmpty()) {
                queue.pop front(&b1,&b2);
                if (isLeaf(b1) ∧ isLeaf(b2))
                        dist ← min(dist,convDist(b1,b2));
                else {
                        HD ← convDist(b1,b2);
                        if (HD > dist)
                                continue;
                        else if (splitRight(b1,b2)) {
                                queue.enqueue(b1, b2→child1, HD);
                                queue.enqueue(b1, b2→child2, HD);
                        }
                        else {
                                queue.enqueue(b1→child1, b2, HD);
                                queue.enqueue(b1→child2, b2, HD);
        }}}
        return dist;
}
```

To perform distance computation between two BVHs, the invention uses a branch-and-bound search algorithm such as that shown just above, which applies to BVHs that are binary trees. At each stage one considers a pair of nodes, one from each tree, as retrieved from a queue of pairs. One begins

by enqueuing the roots and setting the distance *dist* to infinity. If both nodes are leaves, one sets *dist* to their distance if it is smaller than the current value. Otherwise, if the distance between the bounding volumes of the current nodes is no greater than *dist*, then one enqueues two or possibly more (if the BVHs are not binary) pairs of nodes: the current node from one tree paired with each of the children of the current node from the other. The queue ordering policy determines the search order but does not affect the correctness. (For example, a LIFO ordering policy is equivalent to a stack.) *splitRight*(b1, b2) decides which node should be split. *convDist*(b1,b2) computes the distance between two CGPs as mapped by the respective rigid-body transformations.

Those skilled in the art will appreciate that by initializing *dist* to a desired clearance threshold instead of to infinity, one easily modifies the algorithm to more efficiently determine whether there is a clearance of at least *dist* between the volumes swept by the bodies.

For CGPs for which one can compute the support functions over the duration of a screw motion, one apples generalized Gilbert/Foo to conservatively approximate the distance between the volumes swept by those CGPs undergoing screw motion. Observe that if *A, B,* and *C* are convex and $A \cup B \subset C$, then *sweep*(*A*,**x**) $\cup$ *sweep*(*B*,**x**) $\subset$ *sweep*(*C*,**x**): Therefore, the conservative, bounding volume approximation nature of the invention's hierarchies is preserved by taking the convex hulls of swept volumes.

This means that one can do the following to the algorithm of *BVH_Dist*. First, add motions **X** and **Y** (e.g., $\mathbf{X} = (\hat{\mathbf{n}}, \theta, \mathbf{x}_a, \mathbf{x}_{tr})$), which are to be applied to BODY1 and BODY2, to the parameter list. One then generalize the *convDist*(*b*1, *b*2) to a more general function call *convDist*(*b*1, **X**, *b*2, **Y**), which computes the distance between the convex hull of the volume swept by *b*1 under motion **X** and the convex hull of the volume swept by *b*2 under motion **Y**. The resulting method computes a conservative approximation of the distance between the volumes swept by BODY1 and BODY2 under screw motions **X** and **Y**.

Consider the case in which X is a point $\mathbf{x}$:

$$h(\mathbf{w},\mathbf{X}(\mathbf{x})) = \max_{t\in[0,1]}\left\{\mathbf{w}\cdot\begin{pmatrix}\mathbf{R}(\hat{\mathbf{n}},t\theta)(\mathbf{x}-\mathbf{x}_a)\\ +\mathbf{x}_a+t\mathbf{x}_{tr}\end{pmatrix}\right\}. \tag{12}$$

Let $\mathbf{R}_{\hat{\mathbf{n}}z}$ be the minimal rotation that takes $\hat{\mathbf{n}}$ to the z axis. Then

$$\mathbf{R}(\hat{\mathbf{n}},t\theta) = \mathbf{R}^{-1}_{\hat{\mathbf{n}}z}\begin{pmatrix}\cos(t\theta) & -\sin(t\theta) & 0\\ \sin(t\theta) & \sin(t\theta) & 0\\ 0 & 0 & 1\end{pmatrix}\mathbf{R}_{\hat{\mathbf{n}}z}.$$

Defining

$$\begin{aligned}\mathbf{x}^* &= \mathbf{R}_{\hat{\mathbf{n}}z}(\mathbf{x}-\mathbf{x}_a)\text{ and}\\ \mathbf{w}^* &= \mathbf{R}_{\hat{\mathbf{n}}z}\mathbf{w},\end{aligned} \tag{13}$$

and doing some obvious algebra, one has

$$\mathbf{w}\cdot\mathbf{X}(t,\mathbf{x}) = \mathbf{w}^*\cdot\begin{pmatrix}\cos(t\theta) & -\sin(t\theta) & 0\\ \sin(t\theta) & \sin(t\theta) & 0\\ 0 & 0 & 1\end{pmatrix}\mathbf{x}^* + t\mathbf{w}\cdot\mathbf{x}tr + \mathbf{w}\cdot\mathbf{x}a. \tag{14}$$

The $t\in[0,1]$ where $\mathbf{w}\cdot\mathbf{X}(t,\mathbf{x})$ is possibly maximized are 0, 1, and t such that $\frac{\partial}{\partial t}\mathbf{w}\cdot\mathbf{X}(t,\mathbf{x}) = 0$. The last case entails solving for $t\in[0,1]$ such that

$$0 = \frac{\partial}{\partial t}\mathbf{w}\cdot\mathbf{X}(t,\mathbf{x}) = \mathbf{w}^*\cdot\theta\begin{pmatrix}-\sin(t\theta) & -\cos(t\theta) & 0\\ \cos(t\theta) & -\sin(t\theta) & 0\\ 0 & 0 & 1\end{pmatrix}\mathbf{x}^* + \mathbf{w}\cdot\mathbf{x}_{tr}. \tag{15}$$

By inspection, (15) is linear in $\cos(t\theta)$ and $\sin(t\theta)$ and otherwise independent of t, so it can be solved in terms of $t\theta$ by the usual substitution methods, e.g., $y = \sin(t\theta)$, etc. Finding all solutions

$t \in [0,1]$, evaluate $\mathbf{w} \cdot \mathbf{X}(t,\mathbf{x})$ at those t and $t = 0$, 1. The maximizing value is then plugged into (9) and (10) specialized to $\mathcal{X} = \mathbf{x}$.

If $\mathcal{P}$ is a polyhedron with vertices $\mathcal{X} = \{\mathbf{p}_i\}$, one can compute $h(\mathbf{w},\mathbf{X}(\mathcal{P}))$ by computing the values of $h(\mathbf{w},\mathbf{X}(\mathbf{p}_i))$ and choosing the greatest value. Then compute $s(\mathbf{w},\mathbf{X}(\mathcal{P}))$ as $s(\mathbf{w},\mathbf{X}(\mathbf{p}^*))$ for a consistently chosen vertex $\mathbf{p}^*$ that maximizes this value. This enables computation of the minimum distance vector between the convex hulls of the volumes swept by convex polyhedra under screw motion, with accuracy limited only by machine precision and the numerical properties of the algorithm.

This simple method of computing the support function requires $O(n)$ time for a polyhedron with n vertices. To exploit the use of hill-climbing from the enhanced Gilbert/Foo algorithm to efficiently (roughly $O(\sqrt{n}\,)$ computing the support functions for a polyhedron undergoing screw motion requires some care. Assume that which vertices are adjacent to each vertex is pre-computed and stored as a part of the polyhedron representation. This application now describes a technique that incorporates hill-climbing and is roughly $O(\sqrt{n}\,)$ for screw motions with bounded rotation.

Define the support vertex set of a polyhedron $\mathcal{P}$ relative to a vector $\mathbf{w}$ to be the set of all vertices $\mathbf{p}$ such that $\mathbf{w} \cdot \mathbf{p} = h(\mathbf{w},\mathcal{P})$. One can compute $s(\mathbf{w},\mathbf{X}(\mathcal{P}))$ and $h(\mathbf{w},\mathbf{X}(\mathcal{P}))$ for a polyhedron $\mathcal{P}$ undergoing screw motion $\mathbf{X}(t)$; $t \in [0,1]$ by doing the following:

1. Compute the support vertex set of $\mathbf{X}(0,\mathcal{P})$ relative to $\mathbf{w}$, exploiting hill-climbing and taking into account co-planarity of vertices on a common facet.

2. Then, track the changes in the support vertex set as t increases from 0 to 1. For each new vertex $\mathbf{p}$ that enters the set, compute $h(\mathbf{w},\mathbf{X}(\mathcal{P}))$.

3. For a consistently chosen vertex $\mathbf{p}^*$ that maximizes this value, return $s(\mathbf{w},\mathbf{X}(\mathbf{p}^*))$.

To track changes in the support vertex set during screw motion, observe that each vertex that enters the set as t increases must be adjacent (via an edge on the polyhedron) to some current member for the set. Furthermore, at each time t_i when the support vertex set is nondegenerately planar, by taking the dot-product of the axis of rotation $\hat{\mathbf{n}}$ (recall that rotation angles are non-negative) with the cross-products of $\mathbf{w}$ and each member of the current support vertex set, one can determine which members will immediately leave and which will remain until a new vertex is joins the set. The vertices that do not leave the set immediately will be co-linear and lie parallel to the rotation axis. As a consequence of convexity (of the polyhedron), it is sufficient to choose a single vertex $\mathbf{p}_i^*$ from these remaining vertices and determine the smallest time $t_{i+1} > t_i$ at which

$$\mathbf{w} \cdot \mathbf{X}(t_{i+1}, p'_{i+1}) = \mathbf{w} \cdot \mathbf{X}(t_{i+1}, \mathbf{p}_i^*)$$

for some vertex p'_{i+1} adjacent to $\mathbf{p}_i^*$. To get the set of support vertices at time t_{i+1}, apply hill-climbing with plateau-following to $\mathcal{P}$ with respect to $\mathbf{X}^{-1}(t_{i+1}, \mathbf{w})$ using $\mathbf{p}_i^*$ as start.

The key is, given points $\mathbf{x}$ and $\mathbf{y}$, how to find all $t \in [0,1]$ such that

$$\mathbf{w} \cdot \mathbf{X}(t, \mathbf{x}) = \mathbf{w} \cdot \mathbf{X}(t, \mathbf{y}) . \tag{16}$$

Using the definition (3) of screw motion $\mathbf{X}$ one sees that (16) is equivalent to

$$\mathbf{w} \cdot \mathbf{R}(\hat{\mathbf{n}}, t\theta)(\mathbf{x} - \mathbf{y}) = 0 . \tag{17}$$

As above, let $\mathbf{R}_{\hat{\mathbf{n}}z}$ be the minimal rotation that takes $\hat{\mathbf{n}}$ to the z axis. Let

$$\mathbf{w}^* = \mathbf{R}_{\hat{\mathbf{n}}z}\mathbf{w} \quad \text{and} \tag{18}$$

$$\mathbf{x}^* = \mathbf{R}_{\hat{\mathbf{n}}z}(\mathbf{x} - \mathbf{y}). \tag{19}$$

By substitution, condition (17) is true only when

$$\mathbf{w}^* \cdot \begin{pmatrix} \cos(t\theta) & -\sin(t\theta) & 0 \\ \sin(t\theta) & \cos(t\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}^* = 0.$$

By applying the usual trigonometric substitutions, simple algebra, and solving a quadratic equation all solution values of $t \in [0,1]$ can be easily found.

Note that in order to be conservative with respect to numerical precision issues, the current support vertex set at a given t in any given implementation may need to be a superset of the theoretical set at that t.

Now bound the approximation error with respect to the actual distance between the swept volumes, considering contribution to error associated with a single body. By convexity of the CGPs, it suffices to consider the case of a line segment.

First, consider a point rotated by angle θ about an axis a distance r away. The distance between the arc the point follows and the line segment between its original and final locations is bounded by $r\left(1-\cos\left(\frac{\theta}{2}\right)\right)$.

For a line segment that has projected length l perpendicular to the axis of rotation, an $\frac{l}{2}\sin\theta$ term is needed to account for the greatest minimum distance possible between an axis of rotation on the segment and the boundary of the convex hull. For the maximum distance between the region swept by the segment and the convex hull of that region, one obtains the bound.

$$r\left(1-\cos\left(\frac{\theta}{2}\right)\right)+\frac{l}{2}\sin\theta. \tag{20}$$

(20) also bounds the error contribution by a rigid body with CGPs that have maximum projected diameter l. Compared to the $r\sin(\frac{\theta}{2})$ error bound resulting from just computing distance before and after the rotation, one sees that the part of the error dependent on rotational radius r decreases quadratically with the rotation angle instead of linearly by considering the series expansions.

As readily understood by one of skill in the art, the method of the invention can be practiced on control apparatuses for robotic hardware, such as an internal or external microprocessor programmed with software embodying the method of the invention, application specific integrated circuits, and like hardware or hardware/software combinations.

Industrial Applicability:

The invention is illustrated by the following non-limiting examples.

Example 1

In robot motion, such computations are useful to avoid damaging expensive robot hardware and apparatus manipulated by robots in a potentially closed-environment. This approach can be used to determine a clearance, to determine the size of the clearance, and to detect a collision by determining a zero clearance.

Example 2

In virtual reality environments, such computations are useful to maintain realism by preventing objects in the environment from appearing to pass through other objects. The present invention is particularly appropriate for VR applications because of its computational efficiency.

The preceding examples can be repeated with similar success by substituting the generically or specifically described operating conditions of this invention for those used in the preceding examples.

Although the invention has been described in detail with particular reference to these preferred embodiments, other embodiments can achieve the same results. Variations and modifications of the present invention will be obvious to those skilled in the art and it is intended to cover in the appended claims all such modifications and equivalents. The entire disclosures of all references, applications, patents, and publications cited above are hereby incorporated by reference.